UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,632	12/19/2003	Adam J. Simonoff	84734	9927

23501 7590 10/03/2007

NAVAL SURFACE WARFARE CENTER DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17632 DAHLGREN ROAD
SUITE 121
DAHLGREN, VA 22448-5110

EXAMINER	
KHATRI, ANIL	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
10/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

Office Action Summary	Application No.	Applicant(s)	
	10/750,632	SIMONOFF ET AL.	
	Examiner	Art Unit	
	Anil Khatri	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 August 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24 and 26-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24 and 26-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-24 and 26-38 rejected under 35 U.S.C. 102(b) as being anticipated by *Hsu* USPN 5,974,254.

Regarding claims 1, 17 and 34

Hsu teaches,

Providing asynchronous access to multiple users to a graphical programming and analysis environment program (column 4, lines 57-67, he method further comprises a step of matching the first plurality of objects of the first graphical program with the second plurality of objects of the second graphical program. The method attempts to match objects of the graphical programs to determine similarities between the two programs, and hence to aid in finding differences between them. Preferably, the matching is performed according to a matching heuristic which calculates scores indicating a degree of similarity between an object in the first graphical program and an object in the second graphical program according to one or more criteria.

These scores, or matching information, are stored in a match matrix data structure. The rows of the match matrix correspond to the objects of the first graphical program and the columns of the match matrix correspond to the objects of the second graphical program. The matching is performed for both the block diagram graphs and the user interface panel graphs);

allowing each user to generate graphically represented code objects within the environment program (column 7, lines 55-64, FIG. 3, a flowchart illustrating the method of detecting differences between graphical programs according to the present invention is shown. Preferably, the method of the present invention is embodied as a software program which executes on a computer system such as computer system 12 of FIG. 1. The software program of the present invention for detecting differences between graphical programs will subsequently be referred to as "diff" for brevity. Appendix A includes a portion of a C language source code listing of one embodiment of diff);

allowing each user access to the code objects of other users based on security privileges accorded to the user (column 16, lines 39-46 then, diff determines which objects of the user interface panels have no match, in step 164. This occurs when the user has added or deleted an object between the first user interface panel and the second user interface panel. Objects which have no match are identified as those which have a 1:0 relationship in the user interface panel match matrix. The differences found in steps 160 through 164 are stored, in step 166, so that the differences may be displayed in step 112);

allowing each user to have the code objects of the user be chained to the code objects of other users to which the user has access to yield inter-code object communication by inter code object connection each inter-code connection terminating on one of an edge and an interior of one of the code object (column 13, lines 12-21, Using the matching object pairs produced by step 122, diff traverses the graphs of the graphical programs searching for matching edges and increases the score of the source and destination object connected to each matching edge found, in step 124. That is, diff scores the objects by examining the connectivity of the objects. A matching edge is an edge in which both the source and destination objects are elements of matching object pairs according to step 122. According to one embodiment of the matching heuristic, the weight given to objects attached to a matching edge is relatively large; and;

allowing each user to execute application programs composed of the code objects as chained together within the environment program (column 4, lines 25-45, The present invention provides a method for detecting differences between two graphical programs. Preferably, the method is executed on a computer system including a display screen and an input device. In one embodiment, the graphical programs are used to control instruments or other devices coupled to the computer system. The first graphical program comprises a first plurality of objects and the second graphical program comprises a second plurality of objects. The objects include attributes and methods and have a specific object type. Preferably, the objects are represented visually as icons in a user interface panel and/or block diagram. The user interface

panel is used to provide input to and receive output from the graphical program. The block diagram comprises graphical code including user interface panel terminals and function blocks connected by data flow paths, or signals, to perform the desired function of the graphical program. The block diagram receives input for the graphical code from the user interface panel and provides output of the graphical code to the user interface panel. The graphical programs also include attributes themselves).

Regarding claims 2, 15, 16 and 26

Hsu teaches,

providing asynchronous access to the multiple users to the graphical programming and analysis environment program comprises enabling multiple users to log into the environment program remotely, such that the multiple users are able to access the environment program simultaneously (column 9, lines 5-15, In the preferred embodiment, the block diagram comprises a data flow diagram arranged as a directed acyclic graph. The vertices of the graph are the terminals and nodes of the block diagram. The edges of the graph are data path signals which connect the nodes and terminals. The nodes themselves comprise one or more terminals which are connected to the edges. The direction of the edges of the graph is determined by the nodes themselves. For example, if a signal is connected between an output terminal of a first node and an input terminal of a second node, then the direction of data flow on that edge is from the output terminal to the input terminal).

Regarding claims 3 and 35

Rejection of claim 1 is incorporated and further claim recites limitation as in claim 1, therefore claims 3 and 35 are rejected under same rationale.

Regarding claims 4 and 24

Hsu teaches,

allowing each user access to the code objects of the other users based on security privileges accorded to the user comprises rendering visible to each user the code objects of the other users to which the user has access (column 16, lines 39-46 then, diff determines which objects of the user interface panels have no match, in step 164. This occurs when the user has added or deleted an object between the first user interface panel and the second user interface panel. Objects which have no match are identified as those which have a 1:0 relationship in the user interface panel match matrix. The differences found in steps 160 through 164 are stored, in step 166, so that the differences may be displayed in step 112).

Regarding claims 5-6 and 37

Hsu teaches,

allowing each user to have the code objects of the user to be chained to the code objects of the other users to which the user has access comprises allowing the user to graphically chain code objects together, such that a sender object of a pair of graphically chained together code objects is able to send data that are received by a receiver object of the pair (column 13, lines 12-21, Using the matching object pairs produced by step 122, diff traverses the graphs of the graphical

programs searching for matching edges and increases the score of the source and destination object connected to each matching edge found, in step 124. That is, diff scores the objects by examining the connectivity of the objects. A matching edge is an edge in which both the source and destination objects are elements of matching object pairs according to step 122. According to one embodiment of the matching heuristic, the weight given to objects attached to a matching edge is relatively large).

Regarding claims 7, 18-20 and 38

Hsu teaches,

the graphical programming and analysis environment program comprises an applet program, and each c0da object comprises an applet program within a same applet context as the environment program (columns 7-8, lines 65-67 and 1-10, In the preferred embodiment, the graphical programs use graphical data flow programming, such as the LabVIEW graphical programming environment. However, other graphical programming systems may employ the method described herein to detect differences between graphical programs. Examples of systems which may employ the method are Visual Designer from Intelligent Instrumentation, Hewlett-Packard's VEE (Visual Engineering Environment), Snap-Master by HEM Data Corporation, DASYLab by DasyTec, and GFS DiaDem, among others. Programming environments which include graphical programming elements can also use the graphical diff method of the present invention).

Regarding claim 8

Hsu teaches,

at least one of the graphical

programming arid analysis environment program and the code objects is developed within an architecture-independent and Internet web browsing program-independent computer programming technology (column 9, lines 5-15, in the preferred embodiment, the block diagram comprises a data flow diagram arranged as a directed acyclic graph. The vertices of the graph are the terminals and nodes of the block diagram. The edges of the graph are data path signals which connect the nodes and terminals. The nodes themselves comprise one or more terminals which are connected to the edges. The direction of the edges of the graph is determined by the nodes themselves. For example, if a signal is connected between an output terminal of a first node and an input terminal of a second node, then the direction of data flow on that edge is from the output terminal to the input terminal).

Regarding claims 9-12 and 27, 28

Hsu teaches,

the graphically represented code objects coexist with non graphically represented code objects within the environment program (column 9, lines 28-42, Diff then matches objects in the first graphical program with objects in the second graphical program, in step 104. Objects are matched according to one or more criteria, such as object type, connectivity, attributes and position. Preferably, the matching is performed by calculating a weighted score which indicates a degree of matching or similarity between an object in the first graphical program and an object in the second graphical program according to the one or more criteria to produce

matching information. The matching information is used to group the objects into matching subgroups, or sub-graphs, and non-matching sub-graphs for the purpose of determining differences between the two graphical programs. The matching of objects performed in step 104 will be described in more detail with regard to FIG.8).

Regarding claims 13 and 14

Hsu teaches,

the graphical programming and analysis environment program is visually represented as a white board (column 7, lines 27-34, The instruments are coupled to a unit under test (UUT) 23, process or are coupled to receive field signals, typically generated by transducers. The system 10 may be used in a data acquisition and control application, or may instead be used in a test and measurement application. If the system 10 is used in a data acquisition application, the system 10 may also include signal conditioning circuitry 21 coupled between the data acquisition board 20 and transducers).

Regarding claim 21, 36

Hsu teaches,

A data interface indicating first data to be input into the code object and second data to be output by the code object (column 8, lines 57-67, The block diagram is the portion of the graphical program which includes the graphical code to perform the calculations and operations of the graphical program application. The objects in the block diagram include terminals associated with the front panel controls and indicators. The front panel

terminals are used to input and output data between the front panel controls/indicators and the function blocks of the block diagram. The block diagram objects also include function nodes, such as mathematical operators; code execution structures such as for loops, while loops, case statements, and variable references; string functions; file I/O nodes; communication nodes; instrument I/O nodes; and data acquisition nodes, for example. Preferably, the block diagram nodes are connected by data paths, or signals, which determine the flow of data through the block diagram and internal logic to generate the second data from the first data (see abstract).

Regarding claims 22 and 23

Hsu teaches,

each code object has at least one inter-code object communication graphically terminating on one of an edge and an interior of the code object (column 5, lines 7-25, pairs of objects (i.e., one from each of the graphical programs) are scored which match according to object type and which have no conflicts with other objects according to object type. Using these matching objects as starting points, the graphs are traversed looking for matching edges. The score of the source and destination objects of matching edges are updated accordingly. The matrix is then resolved by selecting the highest score in a given row or column. That is, in a given row, for example, the object corresponding to the column element in the match matrix with the highest score is selected to match the object corresponding to the given row. The other elements in the row are then zeroed out to indicate the resolved match. If a tie in scores exists at this point, the tie is not resolved, but postponed until a later step. Thus, when the match matrix is eventually resolved with all ties broken, a 1:1 or 1:0 relationship exists between

objects in the first graphical program and objects in the second graphical program, the relationship depending upon whether or not an object in the first graphical program has a matching object in the second graphical program).

Regarding claims 29-32

Hsu teaches,

one or more application programs are at least one of: capable of being stored for later retrieval and use, and modular ha nature so that more complex application programs may be constructed therefrom (column 9, lines 17-26, in response to receiving the two graphical programs, diff creates a data structure representing the first block diagram, a data structure representing the second block diagram, a data structure representing the first user interface panel, and a data structure representing the second user interface panel, in step 102. Preferably, the data structures comprise directed graphs, and more particularly, directed acyclic graphs. The graphs are used by diff to determine differences between the block diagrams and user interface panels of the two graphical programs. Step 102 and the graph structure will be discussed in more detail with regard to FIG. 6).

Regarding claim 33

Hsu teaches,

a chat area within which the user can communicate with the other users; and,

a user list area showing a name of each of the user and the other users currently logged into the environment program (column 13, lines 12-21, Using the matching object pairs

produced by step 122, diff traverses the graphs of the graphical programs searching for matching edges and increases the score of the source and destination object connected to each matching edge found, in step 124. That is, diff scores the objects by examining the connectivity of the objects. A matching edge is an edge in which both the source and destination objects are elements of matching object pairs according to step 122. According to one embodiment of the matching heuristic, the weight given to objects attached to a matching edge is relatively large).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anil Khatri whose telephone number is 571-272-3725. The examiner can normally be reached on M-F 8:30-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ANIL KHATRI
PRIMARY EXAMINER